UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/610,301	07/05/2000	Derek W. Bolton	CSC01275	4206

29989 7590 03/16/2004

HICKMAN PALERMO TRUONG & BECKER, LLP
1600 WILLOW STREET
SAN JOSE, CA 95125

EXAMINER	
KADING, JOSHUA A	
ART UNIT	**PAPER NUMBER**
2661	8

DATE MAILED: 03/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/610,301	BOLTON, DEREK W.	
	Examiner	Art Unit	
	Joshua Kading	2661	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 January 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21 and 23-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21 and 23-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 July 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Drawings

This application, filed under former 37 CFR 1.60, lacks formal drawings. The informal drawings filed in this application are acceptable for examination purposes. When the application is allowed, applicant will be required to submit new formal drawings. In unusual circumstances, the formal drawings from the abandoned parent application may be transferred by the grant of a petition under 37 CFR 1.182.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1-11, 12-19, 21, 23-32, 33, and 35 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claims 1, 12, 21, 33, and 35 all disclose "retaining...an excess number of duplicate acknowledgements". There is no support for retaining the excess number of duplicate acknowledgements in the specification. Page 11, lines 32-38 of the specification describes the embodiments of applicant's invention. All embodiments describe storing (retaining) the number of duplicate acknowledgements. That is to say,

all embodiments describe storing a count of duplicate acknowledgements, not storing the actual duplicate acknowledgements as described in the above mentioned claims. Page 12, lines 11-38 and page 13, lines 1-31 also only describe saving a count of duplicate acknowledgements. This can be ascertained from the language in the specification. For instance, the count is stored in the variable **excess_dup_ack** and is later compared to variable **dup_ack**, this suggests that both variables only contain a count of duplicate acknowledgements and not the actual duplicate acknowledgements.

It should be noted that if the new matter of claims 1-11, 12-1, 21, 23-32, 33, and 35 is removed, the rejections for claims 1-21, and 23-25 from the previous office action are maintained.

Claim Rejections - 35 USC § 102

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Claim 20 is rejected under 35 U.S.C. 102(b) as being anticipated by RFC2582.

Regarding claim 20, RFC2582 discloses "a transmission control protocol method comprising:

performing a TCP fast recovery process (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", line 1);

performing a TCP fast recover extended process upon receiving acknowledgements of receipt of new data in said TCP fast recover process (section 3

"The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5 where retransmitting the partial segment in response to the partial ACK is an extended packet transmission recovery action).

Claim Rejections - 35 USC § 103

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Taking the amended claims on their merits, claims 1, 3-8, 11-16, 19, and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over RFC2582 in view of Lakshman et al. (U.S. Patent 6,078,564).

Regarding claim 1, RFC2582 dislcoses "a network device-based method comprising:

determining...upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where an excess number of duplicate ACKs is the same as receiving 3 duplicate ACKs because when 3 duplicate ACKs are received, the threshold has been met in excess and the "Fast Recovery procedure" begins); and

taking a network packet transmission recovery action based upon said excess number of duplicate acknowledgements (section 3 "The Fast Retransmit and Fast

Recovery Algorithms in NewReno", lines 8-13 of section 3 when a threshold of duplicate acknowledgements is reached the "Fast Recovery procedure" begins)."

RFC2582 lacks "...retaining..." the duplicate ACKs as they are received. However, Lakshman discloses "...retaining..." the duplicate ACKs as they are received (figure 1 shows a buffer 30 that stores the ACK packets before they are transmitted, it is inherent that the Source of the data packets have an ACK receiving buffer to accommodate the transmission of the ACKs from the Destination buffer 30, this is not only true with the ACKs but also true with the data packets as can be seen with the Source data packet buffer of figure 1 and the Destination packet buffer 97 of figure 2, i.e. the Source has a data packet transmit buffer and the Destination has a data packet receive buffer to accommodate the transmission of packets; since the ACKs are no different, in terms of general packet transmission, they must have the same buffer system).

It would have been obvious to one with ordinary skill in the art at the time of invention to include the retaining of the duplicate ACK packets with the rest of the method for the purpose of accommodating different transmission stream rates. The motivation being that the buffers don't allow packets to be lost because of the difference in stream rates (Lakshman, col. 4, lines 21-34).

Regarding claim 12, RFC2582 discloses "a network device-based method comprising:

determining...upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where an excess number of duplicate ACKs is the same as receiving 3 duplicate ACKs because when 3 duplicate ACKs are received, the threshold has been met in excess and the "Fast Recovery procedure" begins);

deflating a congestion window upon said value of excess number duplicate acknowledgements being less than a transmission control protocol sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 7-13); and

optimizing a size of said congestion window to match a reduction in a quantity of unacknowledged data upon said excess number of duplicate acknowledgements being greater than a transmission control protocol sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 24-31)."

RFC2582 lacks "...retaining..." the duplicate ACKs as they are received. However, Lakshman discloses "...retaining..." the duplicate ACKs as they are received (figure 1 shows a buffer 30 that stores the ACK packets before they are transmitted, it is inherent that the Source of the data packets have an ACK receiving buffer to accommodate the transmission of the ACKs from the Destination buffer 30, this is not only true with the ACKs but also true with the data packets as can be seen with the Source data packet buffer of figure 1 and the Destination packet buffer 97 of figure 2,

i.e. the Source has a data packet transmit buffer and the Destination has a data packet receive buffer to accommodate the transmission of packets; since the ACKs are no different, in terms of general packet transmission, they must have the same buffer system).

It would have been obvious to one with ordinary skill in the art at the time of invention to include the retaining of the duplicate ACK packets with the rest of the method for the purpose of accommodating different transmission stream rates. The motivation being that the buffers don't allow packets to be lost because of the difference in stream rates (Lakshman, col. 4, lines 21-34).

Regarding claims 3 and 13, RFC2582 and Lakshman disclose the methods of claims 1 and 12. Lakshman lacks "deflating a congestion window upon said value of said excess number of duplicate acknowledgements in bytes being less than a number of bytes in a transmission control protocol sender segment." However, RFC2582 further discloses "deflating a congestion window upon said value of said excess number of duplicate acknowledgements in bytes being less than a number of bytes in a transmission control protocol sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 7-13)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the deflating of a congestion window with the methods of claims 1 and 12 for the same reasons and motivation as in claims 1 and 12.

Regarding claim 4, RFC2582 and Lakshman disclose the method of claim 1. Lakshman lacks "optimizing a size of a congestion window to match a reduction in a quantity of unacknowledged data upon said excess number of duplicate acknowledgements being greater than a TCP sender segment." However, RFC2582 further discloses "optimizing a size of a congestion window to match a reduction in a quantity of unacknowledged data upon said excess number of duplicate acknowledgements being greater than a TCP sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 24-31)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the optimizing of a congestion window with the method of claim 1 for the same reasons and motivation as in claim 1.

Regarding claim 5, RFC2582 and Lakshman disclose the method of claim 1. Lakshman lacks "comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold." However, RFC2582 further discloses "comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where it is suggested that the value of 3 is the threshold to which the count is being compared to)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the comparing the excess number of duplicate acknowledgements with the method of claim 1 for the same reasons and motivation as in claim 1.

Regarding claims 6 and 14, RFC2582 and Lakshman disclose the methods of claims 5 and 13. Lakshman lacks "performing a fast retransmit upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is greater than or equal to said duplicate acknowledgement threshold." However, RFC2582 further discloses "performing a fast retransmit upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is greater than or equal to said duplicate acknowledgement threshold (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 3-9 where the Fast Retransmit is part of the recovery method). It would have been obvious to one with ordinary skill in the art at the time of invention to include the performing a fast retransmit with the methods of claims 5 and 13 for the same reasons and motivation as in claims 5 and 13.

Regarding claims 7 and 15, RFC2582 and Lakshman disclose the methods of claims 6 and 14. Lakshman lacks "analyzing a size of a congestion window." However, RFC2582 further discloses "analyzing a size of a congestion window (section 4 "Resetting the Retransmit Timer", lines 15-17 of section 4 where it is implied that the window is analyzed for size to know how many data packets to transmit)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the

analyzing a size of a congestion window with the methods of claims 6 and 14 for the same reasons and motivation as in claims 6 and 14.

Regarding claims 8 and 16, RFC2582 and Lakshman disclose the methods of claims 7 and 15. Lakshman lacks "resizing said congestion window upon said analyzing said size of said congestion window showing said size is greater than a predefined size." However, RFC2582 further discloses "resizing said congestion window upon said analyzing said size of said congestion window showing said size is greater than a predefined size (section 5 "Avoiding Multiple Fast Retransmits, line 14 of section 5 where it is implied the congestion window is reduced after being analyzed)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the resizing of said congestion window with the methods of claims 7 and 15 for the same reasons and motivation as in claims 7 and 15.

Regarding claims 11 and 19, RFC2582 and Lakshman disclose the methods of claims 1 and 12. Lakshman lacks "said method is included in Transmission Control Protocol congestion avoidance." However, RFC2582 further discloses "said method is included in Transmission Control Protocol congestion avoidance (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 1-3 of section 3 where it is the purpose of the fast recovery algorithm to avoid congestion)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the TCP

congestion avoidance with the methods of claims 1 and 12 for the same reasons and motivation as in claims 1 and 12.

Regarding claim 33, RFC2582 discloses "a network device-based method for determining...upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where an excess number of duplicate ACKs is the same as receiving 3 duplicate ACKs because when 3 duplicate ACKs are received, the threshold has been met in excess and the "Fast Recovery procedure" begins); and

taking a network packet transmission recovery action based upon said excess number of duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 when a threshold of duplicate acknowledgements is reached the "Fast Recovery procedure" begins)."

RFC2582 lacks "...retaining..." the duplicate ACKs as they are received. However, Lakshman discloses "...retaining..." the duplicate ACKs as they are received (figure 1 shows a buffer 30 that stores the ACK packets before they are transmitted, it is inherent that the Source of the data packets have an ACK receiving buffer to accommodate the transmission of the ACKs from the Destination buffer 30, this is not only true with the ACKs but also true with the data packets as can be seen with the Source data packet buffer of figure 1 and the Destination packet buffer 97 of figure 2,

i.e. the Source has a data packet transmit buffer and the Destination has a data packet receive buffer to accommodate the transmission of packets; since the ACKs are no different, in terms of general packet transmission, they must have the same buffer system).

RFC2582 and Lakshman lack "a programmable memory including a fast recovery extended method..." Although both RFC2582 and Lakshman explicitly lack a programmable memory storing the method, it would have been obvious to one with ordinary skill in the art at the time of invention to include the programmable memory with the method on it because this is the most efficient and feasible way to implement a method in a computer based communication system and since the method steps cannot exist outside of a medium, there must be some type of programmable memory to write and store the method onto.

It would have been obvious to one with ordinary skill in the art at the time of invention to include the retaining of the duplicate ACK packets with the rest of the method for the purpose of accommodating different transmission stream rates. The motivation being that the buffers don't allow packets to be lost because of the difference in stream rates (Lakshman, col. 4, lines 21-34).

Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over RFC2582 in view of Chapman et al.

Regarding claim 34, RFC2582 discloses "performing a TCP fast recovery process (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", line 1);

performing a TCP fast recover extended process upon receiving acknowledgements of receipt of new data in said TCP fast recover process (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5 where retransmitting the partial segment in response to the partial ACK is an extended packet transmission recovery action).

RFC2582 lacks means for performing the TCP fast recovery process and means for performing a TCP fast recover extended process. However, Chapman further discloses means for performing the TCP fast recovery process and means for performing a TCP fast recover extended process (figure 15 which is a network device that implements all the above tasks).

It would have been obvious to one with ordinary skill in the art at the time of invention to means for performing the TCP fast recovery and means for performing the TCP fast recover extended process. The motivation being that the means of performing these functions allows a user to access these methods and use them in their network communications (Chapman, col. 8, lines 18-26).

Claims 2, 9-10, 17-18, 21-32, 34, and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over RFC2582 and Lakshman as applied to claims 1 and 12 (only for claims 9-10 and 17-18 respectively) above, and further in view of Chapman et al.

Regarding claims 9 and 17, RFC2582 and Lakshman disclose the methods of claims 1 and 12. RFC2582 and Lakshman lack "analyzing a size of a congestion window upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is less than said duplicate acknowledgement threshold." However, Chapman discloses "analyzing a size of a congestion window upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is less than said duplicate acknowledgement threshold (col. 5, lines 33-34 where inflating the window after the receipt of a non-duplicate ACK is received is a method of determining if the window is inflated and this situation occurs when the duplicate acknowledgement threshold is not met)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the analyzing of the congestion window with the methods of claims 1 and 12 for the purpose of controlling flow of packets into the network. The motivation being that by controlling flow into the network congestion and packet loss are reduced to a minimum or tolerable level (Chapman, col. 3, lines 16-30).

Regarding claims 10 and 18, RFC2582, Lakshman, and Chapman disclose the methods of claims 9 and 17. RFC2582 and Lakshman lack "resizing said congestion window upon analyzing said size of said congestion window showing said size is greater than a predefined size." However, Chapman et al. disclose "resizing said

congestion window upon analyzing said size of said congestion window showing said size is greater than a predefined size (col. 8, lines 2-5 where MAX-WND is the predetermined size and C-WND is the actual size of the window)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the resizing of the congestion window with the methods of claim 9 and 17 for the same reasons and motivation as in claims 9 and 17.

Regarding claim 21, RFC2582 discloses "...a fast recovery extended method (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5 where retransmitting the partial segment in response to the partial ACK is an extended packet transmission recovery action)...determine, upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where an excess number of duplicate ACKs is the same as receiving 3 duplicate ACKs because when 3 duplicate ACKs are received, the threshold has been met in excess and the "Fast Recovery procedure" begins)...and take a network packet transmission recovery action based upon said excess number of duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 when a threshold of duplicate acknowledgements is reached the "Fast Recovery procedure" begins)."

RFC2582 lacks "a processor; and a memory coupled to said processor, and storing a fast recovery extended method wherein upon execution of said fast recovery extended method by said processor..." However, Chapman disclose "a processor; and a memory coupled to said processor, and storing a fast recovery extended method wherein upon execution of said fast recovery extended method by said processor a fast recovery process is extended (figure 15, elements 32, 30, and 28; col. 8, lines 27-31 where elements 32 and 28 constitute the processor and 30 is the memory that stores the fast recovery extended method; although the fast recovery extended method is not explicitly disclosed in Chapman, it is suggested that by using the TCP methods of RFC2582 with Chapman, they would need to be stored in the memory in order to be executed)..."

RFC2582 and Chapman further lack to "...retain said excess number of duplicate acknowledgements in said memory..." However, Lakshman discloses "...retain said excess number of duplicate acknowledgements in said memory (figure 1 shows a buffer 30 that stores the ACK packets before they are transmitted, it is inherent that the Source of the data packets have an ACK receiving buffer to accommodate the transmission of the ACKs from the Destination buffer 30, this is not only true with the ACKs but also true with the data packets as can be seen with the Source data packet buffer of figure 1 and the Destination packet buffer 97 of figure 2, i.e. the Source has a data packet transmit buffer and the Destination has a data packet receive buffer to accommodate the transmission of packets; since the ACKs are no different, in terms of general packet transmission, they must have the same buffer system)..."

It would have been obvious to one with ordinary skill in the art at the time of invention to include the fast recovery extended method, the processor and memory, and the retaining of said number of duplicate ACKs for the purpose of implementing the fast recover extended method using the processor and memory, and to accommodate different transmission stream rates. The motivation being that implementing the method using the processor and memory is the only efficient and feasible way possible to do so in a computer based communication system; and the buffers don't allow packets to be lost because of the difference in stream rates (Lakshman, col. 4, lines 21-34).

Regarding claims 2 and 23, RFC2582 and Lakshman disclose the method of claim 1; and RFC2582, Lakshman, and Chapman disclose the method of claim 21. RFC2582 and Lakshman lack "determining whether a congestion window is inflated prior to said determining an excess number of duplicate acknowledgements." However, Chapman discloses "determining whether a congestion window is inflated prior to said determining an excess number of duplicate acknowledgements (col. 5, lines 33-34 where inflating the window after the receipt of a non-duplicate ACK is an indicator that the window is inflated and thus all that is needed to do to determine if the window is inflated is look to see if the last ACK is a non-duplicate). It would have been obvious to one with ordinary skill in the art at the time of invention to include the determining whether a congestion window is inflated with the methods of claims 1 and 21 for the purpose controlling flow of packets into the network. The motivation being that by

controlling flow into the network congestion and packet loss are reduced to a minimum or tolerable level (Chapman, col. 3, lines 16-30).

Regarding claim 24, RFC2582, Lakshman, and Chapman disclose the method of claim 21. Lakshman and Chapman lack "deflating a congestion window upon said value of said excess number of duplicate acknowledgements in bytes being less than a number of bytes in a transmission control protocol sender segment." However, RFC2582 further discloses "deflating a congestion window upon said value of said excess number of duplicate acknowledgements in bytes being less than a number of bytes in a transmission control protocol sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 7-13)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the deflating of a congestion window with the method of claim 21 for the same reasons and motivation as in claim 21.

Regarding claim 25, RFC2582, Lakshman, and Chapman disclose the method of claim 21. Lakshman and Chapman lack "optimizing a size of a congestion window to match a reduction in a quantity of unacknowledged data upon said excess number of duplicate acknowledgements being greater than a TCP sender segment." However, RFC2582 further discloses "optimizing a size of a congestion window to match a reduction in a quantity of unacknowledged data upon said excess number of duplicate acknowledgements being greater than a TCP sender segment (section 3 "The Fast

Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 24-31)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the optimizing of a congestion window with the method of claim 21 for the same reasons and motivation as in claim 21.

Regarding claim 26, RFC2582, Lakshman, and Chapman disclose the method of claim 21. Lakshman and Chapman lack "comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold." However, RFC2582 further discloses "comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where it is suggested that the value of 3 is the threshold to which the count is being compared to)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the comparing the excess number of duplicate acknowledgements with the method of claim 21 for the same reasons and motivation as in claim 21.

Regarding claim 27, RFC2582, Lakshman, and Chapman disclose the method of claim 26. Lakshman and Chapman lack "performing a fast retransmit upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is greater than or equal to said duplicate acknowledgement threshold." However, RFC2582 further discloses "performing a fast retransmit upon said

comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is greater than or equal to said duplicate acknowledgement threshold (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 3-9 where the Fast Retransmit is part of the recovery method). It would have been obvious to one with ordinary skill in the art at the time of invention to include the performing a fast retransmit with the method of claim 26 for the same reasons and motivation as in claim 26.

Regarding claim 28, RFC2582, Lakshman, and Chapman disclose the method of claim 27. Lakshman and Chapman lack "analyzing a size of a congestion window." However, RFC2582 further discloses "analyzing a size of a congestion window (section 4 "Resetting the Retransmit Timer", lines 15-17 of section 4 where it is implied that the window is analyzed for size to know how many data packets to transmit)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the analyzing a size of a congestion window with the method of claim 27 for the same reasons and motivation as in claim 27.

Regarding claim 30, RFC2582, Lakshman, and Chapman disclose the method of claim 21. RFC2582 and Lakshman lack "analyzing a size of a congestion window upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate

acknowledgements is less than said duplicate acknowledgement threshold." However, Chapman discloses "analyzing a size of a congestion window upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is less than said duplicate acknowledgement threshold (col. 5, lines 33-34 where inflating the window after the receipt of a non-duplicate ACK is received is a method of determining if the window is inflated and this situation occurs when the duplicate acknowledgement threshold is not met)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the analyzing of the congestion window with the method of claim 21 for the purpose of controlling flow of packets into the network. The motivation being that by controlling flow into the network congestion and packet loss are reduced to a minimum or tolerable level (Chapman, col. 3, lines 16-30).

Regarding claims 29 and 31, RFC2582, Lakshman, and Chapman disclose the methods of claims 28 and 30. Lakshman and Chapman lack "resizing said congestion window upon said analyzing said size of said congestion window showing said size is greater than a predefined size." However, RFC2582 further discloses "resizing said congestion window upon said analyzing said size of said congestion window showing said size is greater than a predefined size (section 5 "Avoiding Multiple Fast Retransmits, line 14 of section 5 where it is implied the congestion window is reduced after being analyzed)." It would have been obvious to one with ordinary skill in the art at

the time of invention to include the resizing of said congestion window with the methods of claims 28 and 30 for the same reasons and motivation as in claims 28 and 30.

Regarding claim 32, RFC2582, Lakshman, and Chapman disclose the method of claim 21. Lakshman and Chapman lack "said method is included in Transmission Control Protocol congestion avoidance." However, RFC2582 further discloses "said method is included in Transmission Control Protocol congestion avoidance (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 1-3 of section 3 where it is the purpose of the fast recovery algorithm to avoid congestion)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the TCP congestion avoidance with the method of claim 21 for the same reasons and motivation as in claim 21.

Regarding claim 35, RFC2582 "determining, upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where an excess number of duplicate ACKs is the same as receiving 3 duplicate ACKs because when 3 duplicate ACKs are received, the threshold has been met in excess and the "Fast Recovery procedure" begins); and

taking a network packet transmission recovery action based upon said excess number of duplicate acknowledgements (section 3 "The Fast Retransmit and Fast

Recovery Algorithms in NewReno", lines 8-13 of section 3 when a threshold of duplicate acknowledgements is reached the "Fast Recovery procedure" begins)."

RFC2582 lacks means for determining and means for taking recovery action. However, Chapman further discloses means for determining and means for taking recovery action (figure 15 which is a network device that implements all the above tasks).

RFC2582 and Chapman further lack "...retaining..." the duplicate ACKs as they are received. However, Lakshman discloses "...retaining..." the duplicate ACKs as they are received (figure 1 shows a buffer 30 that stores the ACK packets before they are transmitted, it is inherent that the Source of the data packets have an ACK receiving buffer to accommodate the transmission of the ACKs from the Destination buffer 30, this is not only true with the ACKs but also true with the data packets as can be seen with the Source data packet buffer of figure 1 and the Destination packet buffer 97 of figure 2, i.e. the Source has a data packet transmit buffer and the Destination has a data packet receive buffer to accommodate the transmission of packets; since the ACKs are no different, in terms of general packet transmission, they must have the same buffer system).

It would have been obvious to one with ordinary skill in the art at the time of invention to include the means for determining and means for taking a recovery action and the retaining of the duplicate ACK packets with the rest of the network device for the purpose of accommodating different transmission stream rates. The motivation

being that the buffers don't allow packets to be lost because of the difference in stream rates (Lakshman, col. 4, lines 21-34).

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Kalampoukas et al. (U.S. Patent 6,438,101 B1) shows a network device for performing IP/TCP communications and a method for using a window to adapt to network conditions. Qaddoura (U.S. Patent 6,646,987 B1) shows an apparatus and method for performing a TCP recovery process. Mamiya et al. (U.S. Patent Application 2003/0022628 A1) shows TCP system with buffers at both receiver and transmitter ends.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Response to Arguments

The objection to the abstract's length is withdrawn in light of the amended abstract submitted on 5 January 2004.

The objection to the title is withdrawn in light of the altered title submitted on 5 January 2004.

Applicant's arguments with respect to claims 1, 3-8, 11-16, and 19 have been considered but are moot in view of the new ground(s) of rejection. As can be read above, claims 1, 3-8, 11-16, and 19 stand rejected under 35 U.S.C. 112 first paragraph for introducing new matter into the claims. Even if the amended claims 1, 3-8, 11-16, and 19 did not contain new matter, they would stand rejected under 35 U.S.C. 103 as is noted above.

Applicant's arguments filed 5 January 2004 have been fully considered but they are not persuasive.

Applicant argues that for claims 1, 2, 12, and 23, RFC2582 does not disclose "determining...an excess number of duplicate acknowledgements". As can be read in the explanations for the rejections of claims 1 and 12, RFC2582 does in fact disclose "determining...an excess number of duplicate acknowledgements". The excess number is determined by receiving three duplicate acknowledgements. Thus, three duplicate acknowledgements are considered in excess.

Applicant argues for claim 20 that step 5 of RFC2582 is "part of a single fast recovery process" and not itself a separate recovery process. Claim 20 does not claim the "TCP fast recovery extended process" to be a separate process from the "TCP fast recovery process" and therefore step 5 of RFC2582 reads on this limitation of claim 20. As per applicant's own definition of the "TCP fast recovery extended process" (as pointed to in applicant's arguments), step 5 of RFC2582 fully includes the feature of "taking a network packet transmission recovery action based on such an excess number" of duplicate acknowledgements.

Applicant argues that claims 2, 9, 10, 17-18, 21-31, 33, 34, and 35 lack any *prima facie* case of obviousness. Examiner respectfully disagrees and points to the rejections in this office action.

Applicant argues that Chapman for claims 2 and 23 does not disclose the "determining whether a congestion window is inflated prior to deciding whether to

determine said excess number of duplicate acknowledgements". Examiner respectfully disagrees and points to the rejections in this office action.

Applicant argues for claims 9, 17, and 30 that Chapman does not disclose performing a comparison between a count of consecutive duplicate acknowledgements, not an excess number of duplicate acknowledgements, and a threshold". Examiner respectfully disagrees and points to the rejections for claims 9, 17, and 30 in this office action.

Applicant argues that for claim 21, Chapman lacks any mention of the "fast recovery extended method" being stored in the disclosed memory. It is true that Chapman does not disclose this. However, Chapman when combined with RFC2582 fully makes up for Chapman's lacking of the "fast recovery extended method" as can be read in the rejection for claim 21 in this office action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joshua Kading whose telephone number is (703) 305-0342. The examiner can normally be reached on M-F: 8:30AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Douglas Olms can be reached on (703) 305-4703. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Joshua Kading
Examiner
Art Unit 2661

JK
March 11, 2004

KENNETH VANDERPUYE
PRIMARY EXAMINER